Password Generator Python Project

Jared Gibson

School of Cybersecurity, Old Dominion University

CYSE 250: Basic Cyber Programming and Networking

Dr. Shoba Vatsa

April 24, 2023

Password Generator Python Project

Project Proposal

The number 1 issue that permeates every aspect of cybersecurity is the possibility of human error. A proper cybersecurity program is only as strong as its weakest link, and as such, it is important to make sure that every user within any given infrastructure has a strong password as the first line of defense. This project will solve that issue by creating a password that meets Microsoft's official guidelines for creating a strong password:

- At least 12 characters long but 14 or more is better.
- A combination of uppercase letters, lowercase letters, numbers, and symbols.
- Not a word that can be found in a dictionary or the name of a person, character, product, or organization.
- Significantly different from previous passwords.

Hardware Specifications

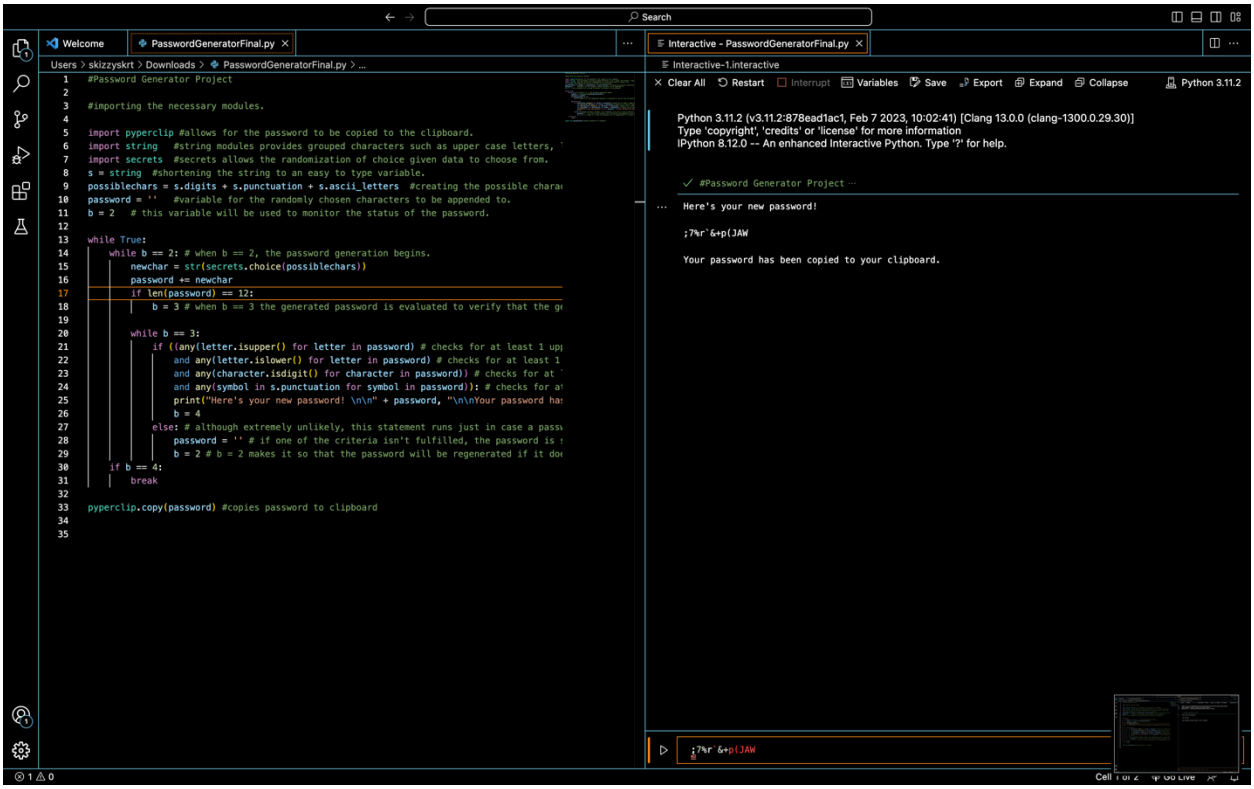
Processor	13th Gen Intel(R) Core(TM) i5-13400F 2.50 GHz
Installed RAM	8.00 GB (7.82 GB usable)
System type	64-bit operating system, x64-based processor

Edition	Windows 11 Home
Version	22H2
Installed on	2/8/2023
OS Build	22621.1555
Experience	Windows Feature Experience Pack 1000.22640.1000.0

The Code

```python
#Password Generator Project

#importing the necessary modules.

import pyperclip #allows for the password to be copied to the clipboard.
import string   #string modules provides grouped characters such as upper case letters, lower case letters, numbers, symbols, etc.
import secrets  #secrets allows the randomization of choice given data to choose from.
s = string  #shortening the string to an easy to type variable.
possiblechars = s.digits + s.punctuation + s.ascii_letters  #creating the possible characters that the generator can choose from. (uppercase letters, lowercase letters, numbers, and symbols.)
password = ''   #variable for the randomly chosen characters to be appended to.
b = 2   # this variable will be used to monitor the status of the password.

while True:
    while b == 2: # when b == 2, the password generation begins.
        newchar = str(secrets.choice(possiblechars))
        password += newchar
        if len(password) == 12:
            b = 3 # when b == 3 the generated password is evaluated to verify that the generated password meets Microsoft criteria.

        while b == 3:
            if ((any(letter.isupper() for letter in password) # checks for at least 1 uppercase letter in the password
                and any(letter.islower() for letter in password) # checks for at least 1 lowercase letter in the password
                and any(character.isdigit() for character in password)) # checks for at least 1 number in the password
                and any(symbol in s.punctuation for symbol in password)): # checks for at least 1 symbol in the password
                print("Here's your new password! \n\n" + password, "\n\nYour password has been copied to your clipboard.") # if all of the above criteria are met, the password is printed, and b == 4
                b = 4
            else: # although extremely unlikely, this statement runs just in case a password is generated without all of the above criteria.
                password = '' # if one of the criteria isn't fulfilled, the password is set to its original empty variable to be regenerated.
                b = 2 # b = 2 makes it so that the password will be regenerated if it doesn't meet Microsoft criteria.
    if b == 4:
        break

pyperclip.copy(password) #copies password to clipboard
```

Successful Run of the Code



Summary of the Code

Prior steps

This code begins by importing a few different modules:

String: Provides us with different lists of characters that we'll be able to use to generate the passwords. We'll be using this to create a list of possible characters to choose from for the password generator.

Secrets: Allows code to choose randoms characters in a given variable. We'll be using this to pick the characters for our password at random.

Pyperclip: This gives us the ability to copy the password to our clipboard once it has been generated and evaluated.

First, we'll create the variable "s" so that it is simpler to call upon the string variable. Next, we'll create our list of possible characters so that we can call upon them to add to our password. In this, we'll be including uppercase letters, lowercase letters, numbers, and non-spacing symbols. We'll then create a variable named "password" to append our randomly chosen characters to. Finally, we'll be creating an arbitrary variable "b" that will monitor the status of the password.

Password Generation

Since b is set to two by default, the password generation process will begin as soon as the code is run. First, we create a variable named "newchar" that will utilize the "secrets" module to choose a random character from our possible characters. Once chosen, it will append that character to the "password" variable. This will repeat until the number of characters within the password is 12. Once that is achieved, b is set to three, sending the password to be evaluated for the Microsoft criteria mentioned before.

Password Evaluation

Our next block of code checks the password against 4 criteria. In order to pass the check, the password must have at least:

- 1 Uppercase Letter
- 1 Lowercase Letter
- 1 Number
- 1 Symbol.

If the code generates a password that meets all four of these criteria, the infinite loop is broken. In the unlikely scenario that the generated password has failed at least one of these criteria, b is set equal to 2 again. This will restart the password generation until a proper password is generated.

Providing the Password

Once out of the loop, the password is printed and copied to the user's clipboard. This will allow them to paste it into any account that they need a password for and save it to their favorite password manager.

References

Bengtsson, T., & Neil. (2018, September 16). *Python - create string with for-loop*. Stack Overflow. Retrieved April 30, 2023, from https://stackoverflow.com/questions/52353982/python-create-string-with-for-loop

JKL_DTEP. (n.d.). *"'bool' object is not iterable" error.* Codecademy. Retrieved April 30, 2023, from https://www.codecademy.com/forum_questions/536a753e9c4e9d8d05000908

M, D., & MarcinMarcin. (2013, July 13). *Typeerror("'Bool' object is not iterable",) when trying to return a boolean*. Stack Overflow. Retrieved April 30, 2023, from https://stackoverflow.com/questions/17630323/typeerrorbool-object-is-not-iterable-when-trying-to-return-a-boolean

manjeet_04. (2023, April 6). *Python: Check if any element in list satisfies a condition*. GeeksforGeeks. Retrieved April 30, 2023, from https://www.geeksforgeeks.org/python-check-if-any-element-in-list-satisfies-a-condition/

Microsoft. (n.d.). *Microsoft*. Microsoft Support. Retrieved April 30, 2023, from https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb

Programiz. (n.d.). *Python for loop*. Programiz. Retrieved April 30, 2023, from https://www.programiz.com/python-programming/for-loop

Python if ... else. W3Schools. (n.d.). Retrieved April 30, 2023, from https://www.w3schools.com/python/python_conditions.asp

Python While Loops. W3Schools. (n.d.). Retrieved April 30, 2023, from https://www.w3schools.com/python/python_while_loops.asp

Python. (n.d.). *Built-in types*. Python documentation. Retrieved April 30, 2023, from https://docs.python.org/3/library/stdtypes.html?highlight=isdigit

Python. (n.d.). *Secrets - generate secure random numbers for managing secrets*. Python documentation. Retrieved April 30, 2023, from https://docs.python.org/3/library/secrets.html

String - common string operations. Python documentation. (n.d.). Retrieved April 30, 2023, from https://docs.python.org/3/library/string.html

Wallen, J. (2023, April 19). *Python for beginners: And/or operators*. The New Stack. Retrieved January 25, 2022, from https://thenewstack.io/python-for-beginners-and-or-operators/

Wilson, H. O. (n.d.). *What are break, continue, and pass statements in python?* Educative. Retrieved April 30, 2023, from https://www.educative.io/answers/what-are-break-continue-and-pass-statements-in-python?utm_campaign=%5BNew%5D%2BPerformance%2BMax&utm_source=adwords&utm_medium=ppc&utm_content=performance_max&eid=5082902844932096&utm_term=&hsa_acc=5451446008&hsa_cam=18511913007&hsa_grp=&hsa_ad=&hsa_src=x&hsa

_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQjwocShBhCOARIs

AFVYq0g9sKrfT8ojiEBvV7VXt1U_MrcMHBoYAxHwYoVc-

P_i1uX2DUZOuYgaAlEdEALw_wcB